

Grading Assistant for Lecturers and Students

Milestone 1

Sean Vowles | 13065858 | CS3D66 | 2016

Table of Contents

[System Requirement Specification 1](#_Toc468476640)

[Introduction 1](#_Toc468476641)

[Purpose of the requirements document 1](#_Toc468476642)

[Scope of the product 1](#_Toc468476643)

[Definitions, Acronyms and Abbreviations 1](#_Toc468476644)

[User Description 1](#_Toc468476645)

[What is Rubrics? 2](#_Toc468476646)

[Example of using rubrics 2](#_Toc468476647)

[Examples of commercial alternatives 5](#_Toc468476648)

[Site Map 6](#_Toc468476649)

[Use Case Description 7](#_Toc468476650)

[Overview: Student 7](#_Toc468476651)

[Overview: Lecturer 7](#_Toc468476652)

[Use Case: Register Account 7](#_Toc468476653)

[System 9](#_Toc468476654)

[Description 9](#_Toc468476655)

[Front End 9](#_Toc468476656)

[Platforms 10](#_Toc468476657)

[Marking Assistant Platforms and Justification 11](#_Toc468476658)

[Server 11](#_Toc468476659)

[Environment 13](#_Toc468476660)

[Classes 14](#_Toc468476661)

[Dashboard 14](#_Toc468476662)

[Users 14](#_Toc468476663)

[Student 14](#_Toc468476664)

[Lecturer 14](#_Toc468476665)

[Assignments 14](#_Toc468476666)

[Grade 14](#_Toc468476667)

[Initial Class Diagram 15](#_Toc468476668)

[Assumptions: 15](#_Toc468476669)

[Legal Social Ethical and Professional Issues 16](#_Toc468476670)

[Legal Aspects: 16](#_Toc468476671)

[Licences for software? 16](#_Toc468476672)

[Types of software licences 16](#_Toc468476673)

[Who owns the project? 17](#_Toc468476674)

[Ethics of software produced 17](#_Toc468476675)

[What is ethics? 17](#_Toc468476676)

[Ethics in computer software and web applications 17](#_Toc468476677)

[Difference between ethics and morals 18](#_Toc468476678)

[Bibliography 19](#_Toc468476679)

[Appendices 21](#_Toc468476680)

# System Requirement Specification

## Introduction

The readers of this document are expected to be lecturers at a University looking to improve the speed and accuracy of assignments that need to be marked.

The system described in this document is intended to be used to describe the process of analysing and designing an object-oriented software solution. The system will be to automate marking and grading of student’s assignments using a web driven application.

## Purpose of the requirements document

This document is to be used to describe the design techniques involved with developing the system. The document provides the requirements of the web application. It will be supplemented by Use Case specifications, Site Map and Class Diagram.

## Scope of the product

The aim of the project is to create a marking assistant for lecturers. The software should assist lecturers in grading coursework and providing feedback for students and their submissions. The lecturers should be able to create, rate and insert comments, create, save and use rubrics which automatically rescale, total marks and convert the mark to percentages, weighted marks and grades. The feedback should be generated automatically based on the grading obtained for individual assignment tasks. The software should be implemented using the principles of OO design, design patterns and persistent storage.

## Definitions, Acronyms and Abbreviations

UML - Unified Modelling Language

OOP - Object Oriented Programming

PHP - Hypertext Pre-processor

SQL - Structured Query Language

SASS - Syntactically Awesome Style Sheets

SCSS - Sassy Cascading Style Sheet

CSS - Cascading Style Sheets

CRUD - Create, Read, Update, Delete

IIS - Internet Information Service

## User Description

1. Present user with login and registration
2. Present user with a list of submitted assignments
3. Provide a list of marked and unmarked assignments
4. Provide generated marking scheme
5. Generate feedback based on grades achieved
6. Enable users to upload assignment
7. Allow users to update passwords of login

# What is Rubrics?

“A rubric is a coherent set of criteria for students’ work that includes descriptions of levels of performance quality on the criteria” (Brookhart, 2013).

Rubrics works by dividing an assignment into parts of sections that are provided with a clear description of the tasks that need to be completed (University C, 2016).

The advantages of using a rubric is that the student is being graded to a specific criteria and that the lecturer also must mark to that criteria. This promotes all round fairness between students and prevents biased marking. Grading rubrics on large courses where there may potentially be more than one person marking an assignment allows for grading consistency.

On a smaller scale, rubrics can also help tutors and students get a clearer picture of the student’s strengths and weaknesses. On a larger scale, a lecturer will be able to identify an average mark across a set of students showing the lecturer an area of teaching that needs more attention.

A rubric is typically documented in table format. The table will have a list of criteria and relevant marking scheme which reads from left to right. Once a column on the left has been achieved the student can then move onto the next column. When developing a software solution, it can typically be assumed that if the grade on the left most has not been achieved, it’s fair to assume the student hasn’t met the minimum requirements to pass as rubric’s works on incremental grading.

## Example of using rubrics

Table A:

Example rubric of the game “Tic-Tac-Toe”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Fail (<40) | Pass (40 – 59) | Merit (60 – 69) | Distinction (70+) |
| Use of arrays, variables and function names  / 10 total marks | Missing or demonstrating little understanding of this technique. | Showing basic knowledge and understanding of this technique. | Shows moderate understanding of arrays. | Clearly understands and well implemented array. Data types and functions have meaningful names. |
| Logic and structure  / 10 total marks | Game doesn’t work or follow any of the rules. | Basic functionality of the game has been implemented. | Game works fluidly and detects winners and losers. | Game works fluidly, saves scores and games won by each player. |
| Use of classes and functions  / 55 total marks | No classes or functions have been attempted | Little use of functions. Game works but lots of redundant code. | Good use of functions and shows moderate understanding of classes. | Excellent understanding and implementation of using classes and functions and applying design patterns. |
| Report  / 25 total marks | Poor report structure, hand drawn diagrams and no referencing. | Report structure unclear, diagrams could be clearer. Diagrams and documentation have inconsistencies. Spelling mistakes in report. | Report structure clearly reflects assessment criteria. Documentation is clear, diagrams and documentation and few inconsistencies. | Report well formulated. Diagrams are clear and neatly produced. Diagrams and documentation have very few or no inconsistencies. |

Table B:

Example rubric from Advanced Databases and Modelling Year 3 Assignment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Fail (<40) | Pass (40 – 59) | Merit (60 – 69) | Distinction (70+) |
| Primary / Foreign Keys  / 20 total marks | Missing or demonstrating little understanding of this technique. | Identifying primary keys and foreign keys has been attempted, but a significant number are missing, or contain significant errors. | Most appropriate primary keys and foreign keys identified. Contains some errors. | All appropriate primary keys and foreign keys identified with no errors. |
| Local Data Structure  / 35 total marks | Missing or demonstrating little understanding of this technique. Model fails to represent case study and/or requires extensive corrections. | Model loosely represents the case study. Reasonable use of notation but contains some major errors or several small errors. Adequate attempt at rationalising M:M relationships (if needed). | Model represents the main aspects of the case study. Most entity types & relationships included. Good use of notation and contains on a few errors. Good attempt at rationalising M:M relationships (if needed). | Model clearly represents the case study. All appropriate entity types & relationships included. Excellent use of notation and very few errors. Clear & correct rationalising of M:M relationships (if needed). |
| Assumptions  / 10 total marks | Missing or inappropriate assumptions. | Some reasonable assumptions made. | A good set of assumptions, mostly appropriate. | An excellent set of appropriate assumptions. |
| Normalisation  / 25 total marks | Little/no demonstration or understanding normalisation. Missing or poor attempt containing many errors. | Normalisation process attempted. Contains errors, and/or documentation is incomplete. | Normalisation process attempted and documented. Normalisation attempted to 3NF, but contains errors. | Demonstrates excellent understanding of normalisation. Process clearly & accurately documented. All screens/forms fully normalised to 3NF. |

(Boobyer, 2016)

The instructional rubric in Table A and Table B both have to components of similarity; they both have criterion to meet and a marking scheme to follow. Using the criteria and marking scheme that is well written and documented allows students to identify weaknesses in their work.

## Examples of commercial alternatives

The technique of using rubrics to describe a marking scheme of an assignment is becoming more and more popular in the twenty-first century of teaching. It is important that software houses keep up with the trend developing applications that support academia and presenting various types of programs from word plugins to standalone applications. There is currently a handful of powerful rubric creation tools available to assist lecturers and teachers producing a well-documented digital evaluation rubric for their students. However, software that assists the lecturers and teachers in marking is particularly scarce with only one reputable software platform available called ‘Rubric-O-Matic’. Rubric-O-Matic allows the use of rescaling marks by highlighting and colouring a cell in the rubric table. Rubric-O-Matic is an extension for Microsoft Word meaning the user needs a licence and an operating system that supports this product. (eMarking Assistant, 2016).

Currently there does not appear to be software available that creates and marks rubrics which also dynamically saves data to persistent storage, is available cross platform or as a web interface.

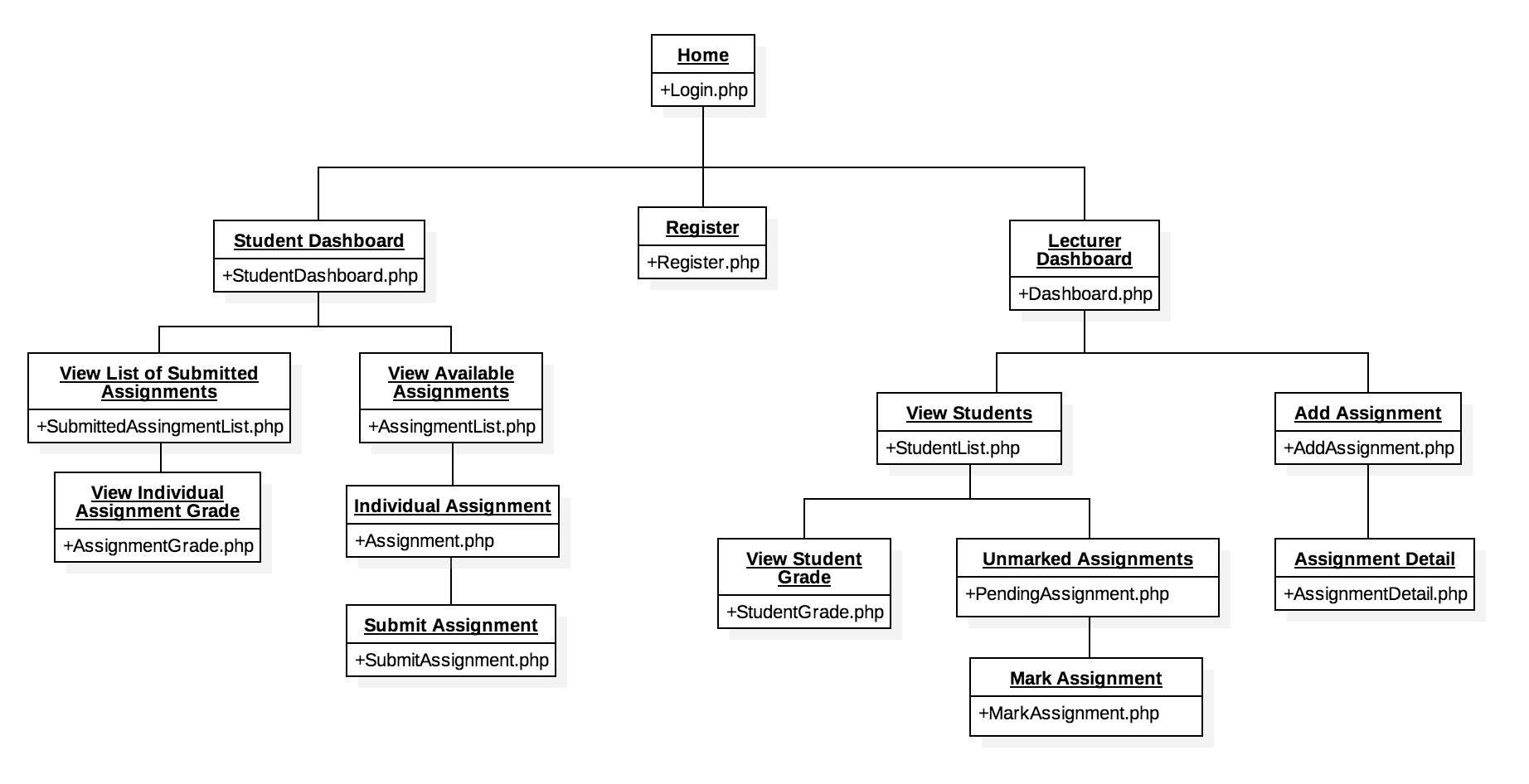
Rubric-O-Matic has additional features available follows:

* Penalties and bonuses
* Total marks
* Percentage conversion
* Assignment weight

Rubric-O-Matic doesn’t support:

* Weighted grade across multiple assignments
* Ability to insert comments
* Provide overall feedback

# Site Map



A web application needs a strict set of pages for the users to create, read update and delete the desired content or information. Using the above site map, it is illustrated that a lecturer and a student will have two separate dashboards. Using this approach, each user can have a list of links or sections that serves their purpose therefore hiding any unnecessary content from the front end. Even though this is a CRUD application it is fair to assume only a lecturer or system administrator will have the privileges to delete any content. It is also a fair assumption that based on the nature of application a student type user should only be able to read content. A student shouldn’t be able to delete any assignments, remove or edit other students from the database, delete unmarked assignments, edit unmarked and submitted assignments, add an assignment or view other student’s records. See page (x) for legal and ethical views on this.

# Use Case Description

## Overview: Student

1. System prompts for user details
2. User inputs details
3. System checks if account exists
4. [Extension: Register Account]
5. System will check for incorrect login
6. System will show available assignments
7. User will choose available assignments
8. User will submit assignment
9. System will save assignments to database
10. User will check grades
11. System will present grades
12. User will view feedback
13. System will present feedback

**Extension Points:**

Register Account: User must register an account

## Overview: Lecturer

1. System prompts for user details
2. User inputs details
3. System checks if account exists
4. [Extension: Register Account]
5. System will check for incorrect login
6. System will prompt to add an assignment
7. User will add assignment
8. System will prompt for a student
9. User will select a student
10. System will prompt for assignment
11. User will choose assignment
12. User will mark assignment
13. System will add marks to database

**Extension Points:**

Register Account: User must register an account

## Use Case: Register Account

1. System generates a new ID
2. System prompts for email
3. User inputs email
4. System prompts for password
5. User inputs password
6. System checks credentials don’t already exist
7. System stores the new records

(See Appendix A)

# System

## Description

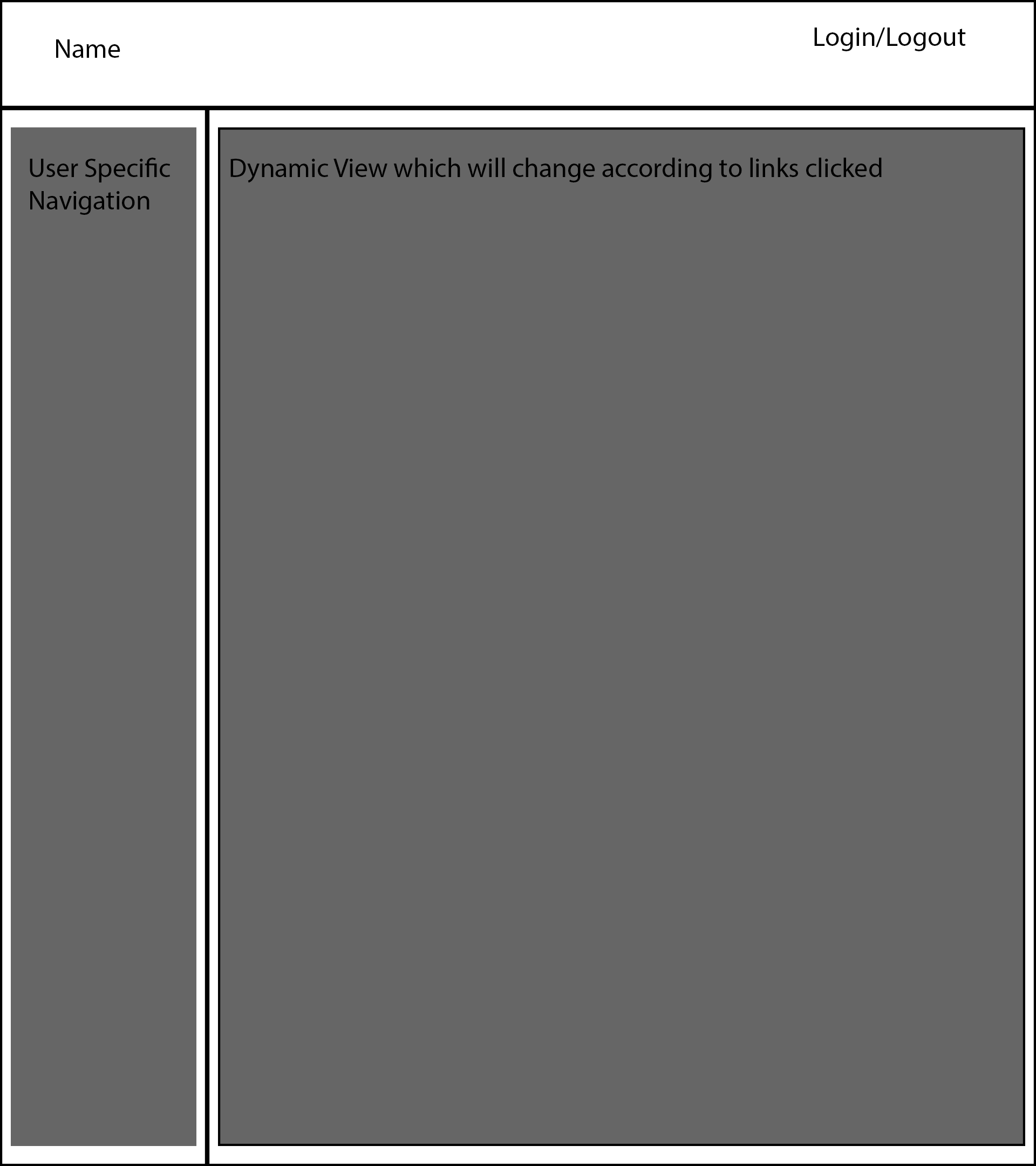
A user will log into the marking assistant application. The application will check if the user is a student or a lecturer and present the user with the relevant dashboard. A dashboard is the homepage for a user and presents relevant information for the type of user checked. A lecturer will be able to add an assignment, view all students, check which students have submitted their assignments, mark said assignment and view all student grades to track their performance. A student will see which assignments have been released. From there a student can download or view an assignment, submit an assignment and check their overall and individual assignment grades.

## Front End

The system front end will be generated using PHP template files. The template files will be ‘header.php’, ‘sidebar.php’ and ‘body.php’.

The ‘sidebar.php’ will be connected to a database and will populate each navigation link through a for each loop iterating through the rows of a student navigation table or a lecture navigation table.

The ‘body.php’ file will be an empty placeholder for the dynamic view of each side bar item.



# Platforms

(Acunetix, 2016) states that a website application is a software solution that allows users to retrieve data and interact with content within pages within a website.

For a software solution to be considered a website application it must have three components that interact with each other; a client side browser and computer, a server side system and a database server.

The client will send a HTTP/S requests to the web server. The web server response will send HTML, CSS, data, JavaScript and images). A database server holds the database engine such as SQL Server, Oracle, MongoDB. This information is sent up to the web server, which is then passed to client if requested. (See appendix B).

The client will view a web application through a browser; this may be Internet Explorer, Edge, Google Chrome, Safari, Opera, Mozilla Firefox, iOS Safari, Android Browser or one of the many browsers that are available today.

Unfortunately, a web application may behave as desired in one browser, but may have issues when run in another browser which could be vital to the application working. There are many components involved with web applications that can produce different results such as CSS, JavaScript, jQuery and Ajax requests. (Segue Technologies, 2016).

When developing a website application, the developer must consider the target audience. The software developer is either going to be building a solution that is going to be available on the public domain or the software being developed is going to be for an intranet (a local or restricted communications network created using World Wide Web Software) of the company purchasing the software. When developing software specific for a company’s intranet, you will have rigid specifications and will typically be told what website browser you will be developing for. This is important as large company’s often use older operating systems and web browsers; especially in the public sector. When developing web applications that are targeted for the public domain, typically, applications will need to be supported on a variety of client browsers. This is a time consuming and crucial part of the development and testing phase. Using W3schools you can identify the most popular browsers and make assumptions based on popularity which browsers need to be supported. (See appendix C)

The main purpose of the web server is to store the files for the website application which is then served back to the client as a response from the client request and to send data between the data server. A website server is assigned an IP address which is how the data is accessed.

A data server is the central point of persistent storage of any information relating to an application and can be stored in either relational or non-relational databases. Relational and non-relational databases are not a substitute for one another rather chosen on the needs of the application being developed. SQL and relational databases are excellent for representing and working with sets of data such as dealing with user accounts of a banking system. SQL and relational databases have tight rules thus also ensuring data security. However, SQL and relational databases fall short when it comes to scaling. The larger and more complicated a dataset becomes, the slower the database will iterate and query. Data servers have two types of scaling; vertical and horizontal. Vertical scaling is increasing the performance of an individual server, horizontal scaling is spreading the data across and adding multiple servers but at the cost of security for the application. (Upguard.com, 2016).

An example of a non-relational database is ‘MondoDB’ which is also defined as a NoSQL database. NoSQL databases use the concept of key-value pairs which are created and stored as BSON which is a binary form of representing data called objects or documents.

# Marking Assistant Platforms and Justification

## Server

The marking assistant for lecturers is going to be hosted on a virtual machine in a cloud environment. The cloud environment of choice is going to be Amazon Web Services. Amazon Web Services offer secure platforms, database storage, content delivery and a free one year subscription of up to 750 computing hours per month.

Initially, a virtual machine will be launched using the Amazon Web Services control panel. Amazon provides a list of stable bootable operating systems that can be installed on their virtual machines using one click.

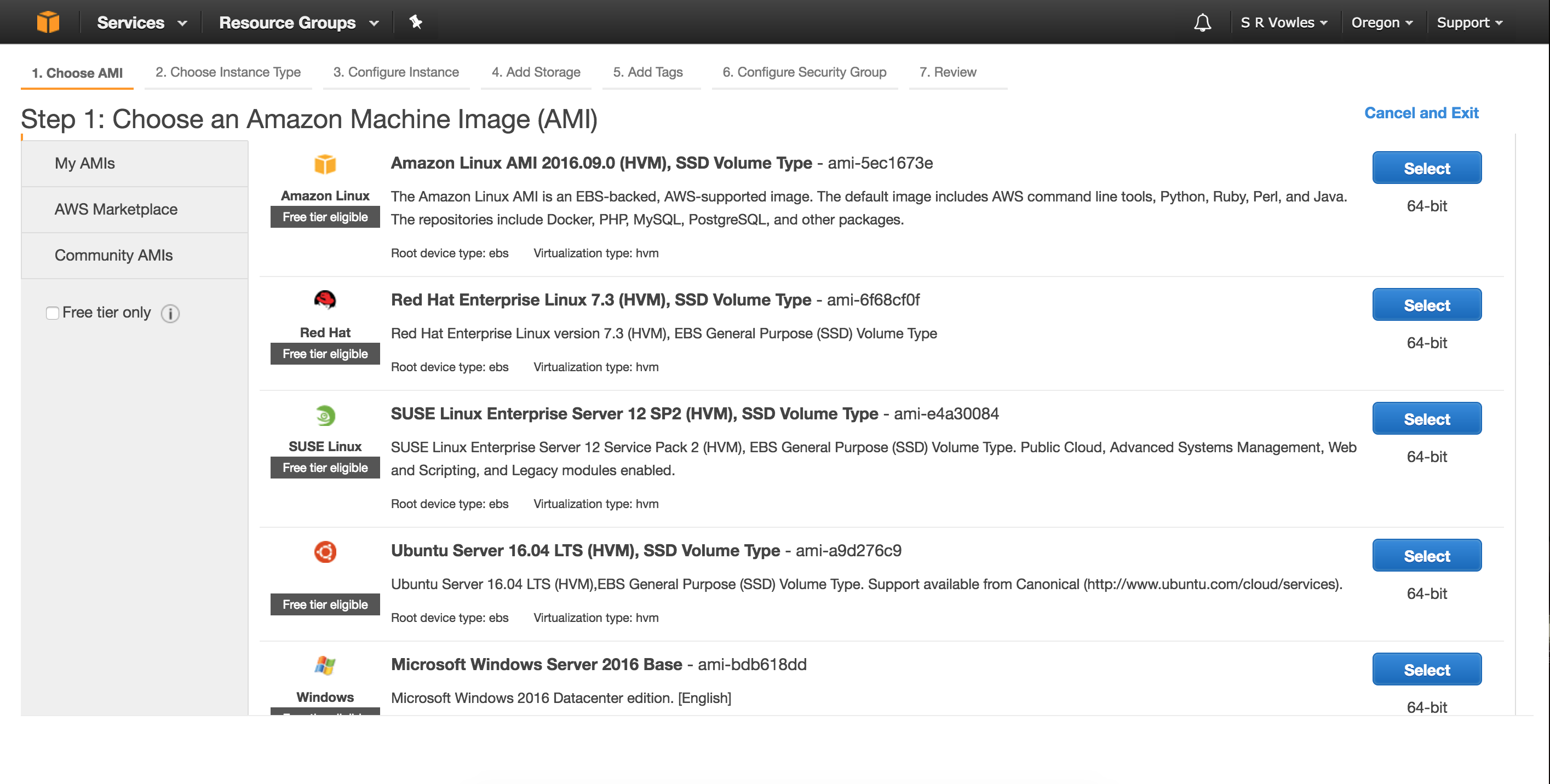


Figure 1

Once an operating system has been selected the user will be directed to the instance configurations. The user will be able to choose from a list of instances which will show how many CPU’s the instance has, the processor type, physical memory, storage space and network performance. A user can choose a basic instance ‘t2.micro’ for initial development and deployment and scale up the system to meet their requirements later.

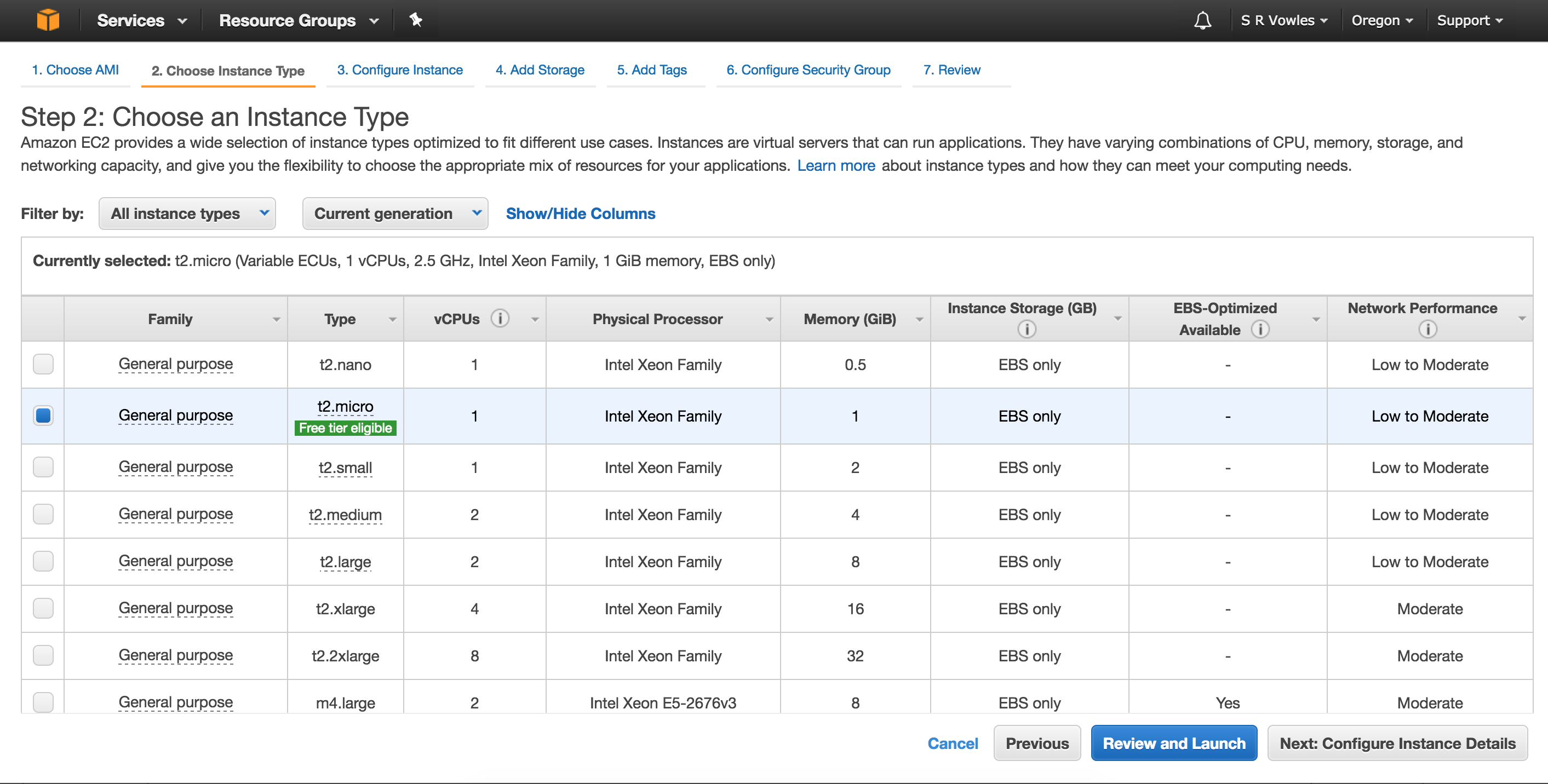


Figure 2

Once an instance is successfully set up it will be operating with a clean operating system installation. For the marking assistant, the operating system of choice is Ubuntu Server, which is a Linux based operating system with only a command line interface. For less experiences users, Ubuntu server can have a desktop environment installed on top and be accessed using remote desktop connection server.

For this project, the operating system will be left at its default and only allowing SSH connections. For this, anyone who needs access to the server must request an SSH key from the system administrator.

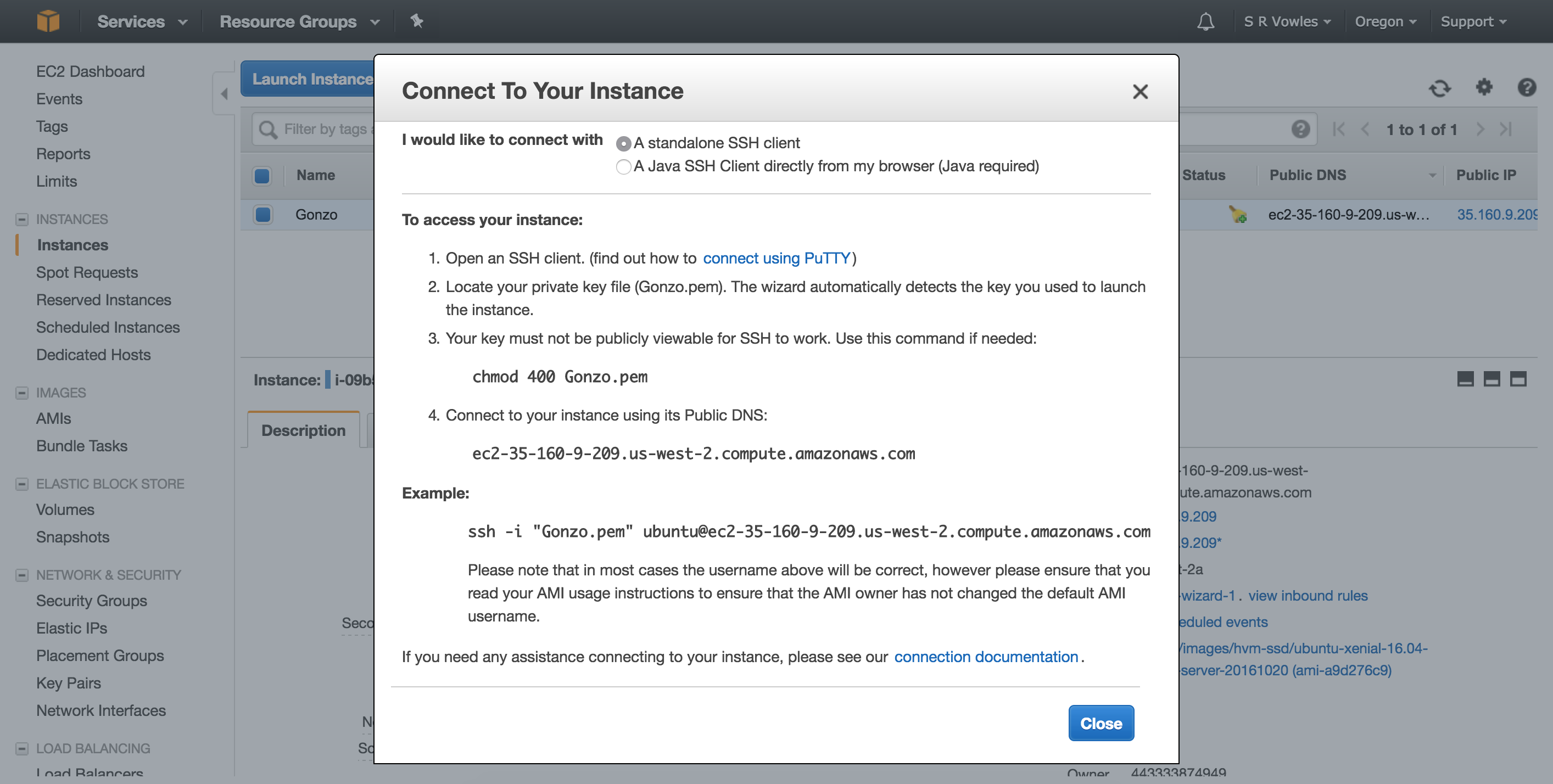


Figure 3

## Environment

For a complete web application solution, the developer requires a stack to run its services. A stack consists of a computer with an operating system, a web server to handle the requests and responses of the web files, persistent storage to store any information or data that is crucial to the application and a language that handles the logic of the application.

There are a variety of different stacks available for web applications ranging from full JavaScript stacks using node.js as the server and AngularJS as the logic, C-sharp’s .NET framework with IIS server or LAMP stack.

The marking assistant for lecturers is going to be developed using a Linux operating system, apache web server, MySQL database and PHP; this is commonly referred to as the LAMP stack. The reasons for using a LAMP stack because all the software is open source and integrates well with each other.

# Classes

### Dashboard

* Visual representation or ‘view’ of data
* Generates menu

### Users

* Holds user information
* Registers user
* Logs in user
* Logs out user
* Checks the user’s role

### Student

* Has a grade
* View’s assignment
* Submit assignment

### Lecturer

* View list of students
* Upload assignment
* Mark assignment

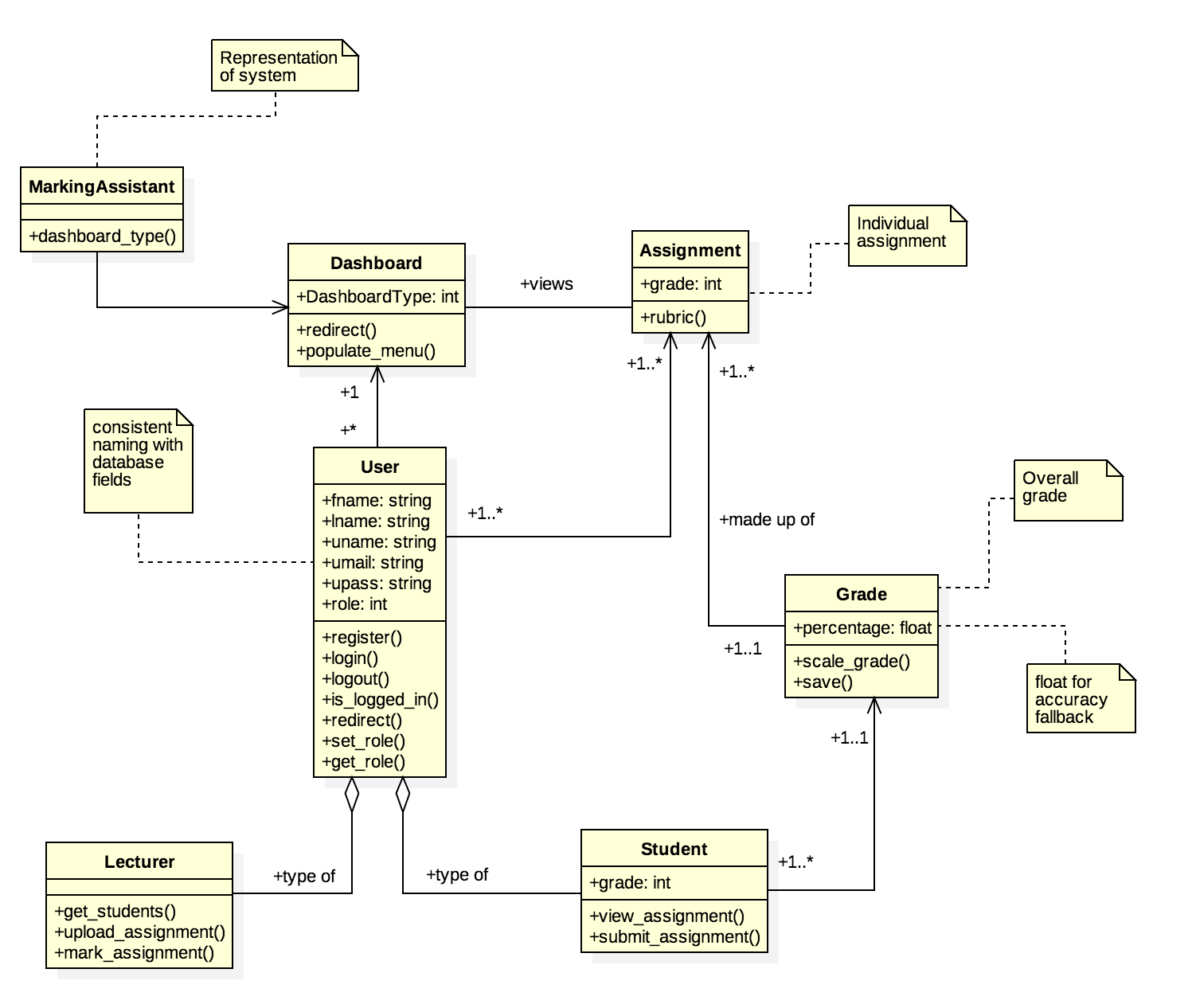
### Assignments

* Holds assignment information
* Has a grade
* Has marking criteria

### Grade

* Percentage
* Scales grade
* Save grade

## Initial Class Diagram



### Assumptions:

The grade class will initially save the percentage as floats. This will allow for the division of odd numbers and keep accuracy. This will then be cast to an integer in the student class.

Overall grade will be made up of many assignments.

Lecturers will have direct access to the students, students will only have access to themselves.

As suggested by Goodrich, an assignment shouldn’t have too little or too much criteria’s. This system is going to be strict allowing only 4 criteria’s. The criteria will be marked out of a total of 100. The lecturer will be able to change the amount of marks available for each criterion.

# Legal Social Ethical and Professional Issues

## Legal Aspects:

### Licences for software?

A software licence is a legally binding agreement that specifies the terms of use of the application for the end user. All software must be legally licenced before it may be installed. (Its.uncg.edu, 2016)

### Types of software licences

The university of North Carolina Greensboro states there are nine different types of software licences available.

#### Propriety Licences

Ownership of the software belongs to the software publisher. The software publisher grants a licence to **use** one or more copies of the software.

#### GNU General Public Licence

General agreements where many open source projects are licenced. Software licenced under GNU allow edits to the source code. However, a change in source code must also be made under the GNU licence, meaning that a piece of open source software cannot be used, changed and then sold. The software or changes to the software cannot be made distributed for a fee.

#### End User Licence Agreement (EULA)

The end user licence agreement indicates the terms in which an end user may use a piece of software. End user licence agreements can be typically seen with subscription based software.

#### Workstation Licence

A type of licence that is only allowed to be installed on one machine. You can only install on multiple machines by purchasing multiple licences. Typically, under a workstation licence you are allowed to make a single backup of the product as long as it is used on the same machine.

#### Concurrent Use Licence

Concurrent use licence allows the installation of a product on any number of machines, however, the software cannot be running on more machines that the amount of licences owned.

#### Site Licence

Allows the use of a piece of software on any computer on a specified site. Unlimited site licences allow the installation of software on any number of computers on the site.

#### Perpetual Licences

Most software installed on people’s home machines fall under the perpetual licence category. A perpetual licence permits the use of software indefinitely as they do not have expiration dates.

#### Non-Perpetual Licences

The user pays to use software over a period of time. If the user stops paying the licence fee they are required to remove the software.

#### Licence with Maintenance

Software sold with a maintenance licence allows the user to upgrade and receive updates of the software until the maintenance agreement expires.

### Who owns the project?

The intellectual property of the marking assistant project remains the students although it is supervised by a lecturer at the faculty.

## Ethics of software produced

### What is ethics?

As suggested by Santa Clara University ethics can be divided into two things; firstly, it refers to well-founded standards of right and wrong. Ethics are built up of rational examinations into people’s moral beliefs and behaviours. Examples of ethics would be the obligations to refrain from rape, stealing, murder and assault. Ethics serve as a moral guide in everyday life.

Secondly is the study of individual’s ethical standards. It is necessary to constantly examine the standards of individual’s ethical beliefs in order to ensure they are reasonable and well-founded. The continuous effort of studying the beliefs of others allows us to live up to standards that are reasonable.

### Ethics in computer software and web applications

Ethics in software and web driven application is primarily focused on the security of data and personal information. The UK Data Archive suggests that the publisher or developer of software has a duty of confidentiality towards information and participants; a duty to protect participants from harm, by not disclosing sensitive information. Individuals using software should have the right to choose whether their information may be used, processed or shared.

Software developers and professionals in the I.T. industry have a professional board to follow known as the British Computer Society. The BCS has the responsibility to set rules and standards to direct the behaviour of its members.

### Difference between ethics and morals

Ethics are a set of rules and boundaries established by society by professionals that we should follow and are generally uniform from society to society, whereas morals are inner beliefs of the individual which are not set by society, and if used to follow principles may cause offence to others. Morals typically differ from society to society and culture to culture (S, 2016).

Moral beliefs can be beliefs to an individual or society that has been learned from elders which are formed from opinion, personal upbringing, cultural influence and comprehension of religious and spiritual principles. (Reference, 2016).

As ethics are governed and established by professional bodies, religious and spiritual beliefs are often not considered when it comes to determining rules as this can cause conflicts between societies and cultures; instead data is collected from a large collection of individuals, cultures and societies ensuring we live to rules that are well founded, just and reasonable.

# Bibliography

Acunetix. (2016). *What Are Web Applications?*. [online] Available at: http://www.acunetix.com/websitesecurity/web-applications/ [Accessed 2 Dec. 2016].

Boobyer, G. (2016). *IS3S602*. [online] Blackboard. Available at: http://www.blackboard.southwales.ac.uk [Accessed 29 Nov. 2016].

Brookhart, S. (2013). *How to create and use rubrics for formative assessment and grading*. 1st ed. Alexandria, VA: ASCD.

Data-archive.ac.uk. (2016). *UK Data Archive - ETHICAL / LEGAL*. [online] Available at: http://www.data-archive.ac.uk/create-manage/consent-ethics/legal [Accessed 1 Dec. 2016].

eMarking Assistant. (2016). *Rubric-O-Matic - automated rubric maker*. [online] Available at: http://emarkingassistant.com/products/rubric-o-matic/ [Accessed 21 Nov. 2016].

Goodrich Andrade, H. (2016). Using rubrics to promote thinking and learning. *Educational Leadership*, 57(5), pp.13-18.

Its.uncg.edu. (2016). *Software Licensing, Information Technology Services, UNCG*. [online] Available at: https://its.uncg.edu/Software/Licensing/ [Accessed 1 Dec. 2016].

Reference. (2016). *What are examples of morals?*. [online] Available at: https://www.reference.com/world-view/examples-morals-5473b4217c1e38d [Accessed 2 Dec. 2016].

S, S. (2016). *Difference Between Morals and Ethics (with Comparison Chart) - Key Differences*. [online] Key Differences. Available at: http://keydifferences.com/difference-between-morals-and-ethics.html [Accessed 2 Dec. 2016].

Segue Technologies. (2016). *The Importance of Cross Browser Testing*. [online] Available at: http://www.seguetech.com/importance-cross-browser-testing/ [Accessed 2 Dec. 2016].

University, C. (2016). *Rubrics-Teaching Excellence & Educational Innovation - Carnegie Mellon University*. [online] Cmu.edu. Available at: https://www.cmu.edu/teaching/designteach/teach/rubrics.html [Accessed 26 Nov. 2016].

University, S. (2016). *What is Ethics? - Ethical Decision Making - Ethics Resources - Markkula Center for Applied Ethics - Santa Clara University*. [online] Scu.edu. Available at: https://www.scu.edu/ethics/ethics-resources/ethical-decision-making/what-is-ethics/ [Accessed 1 Dec. 2016].

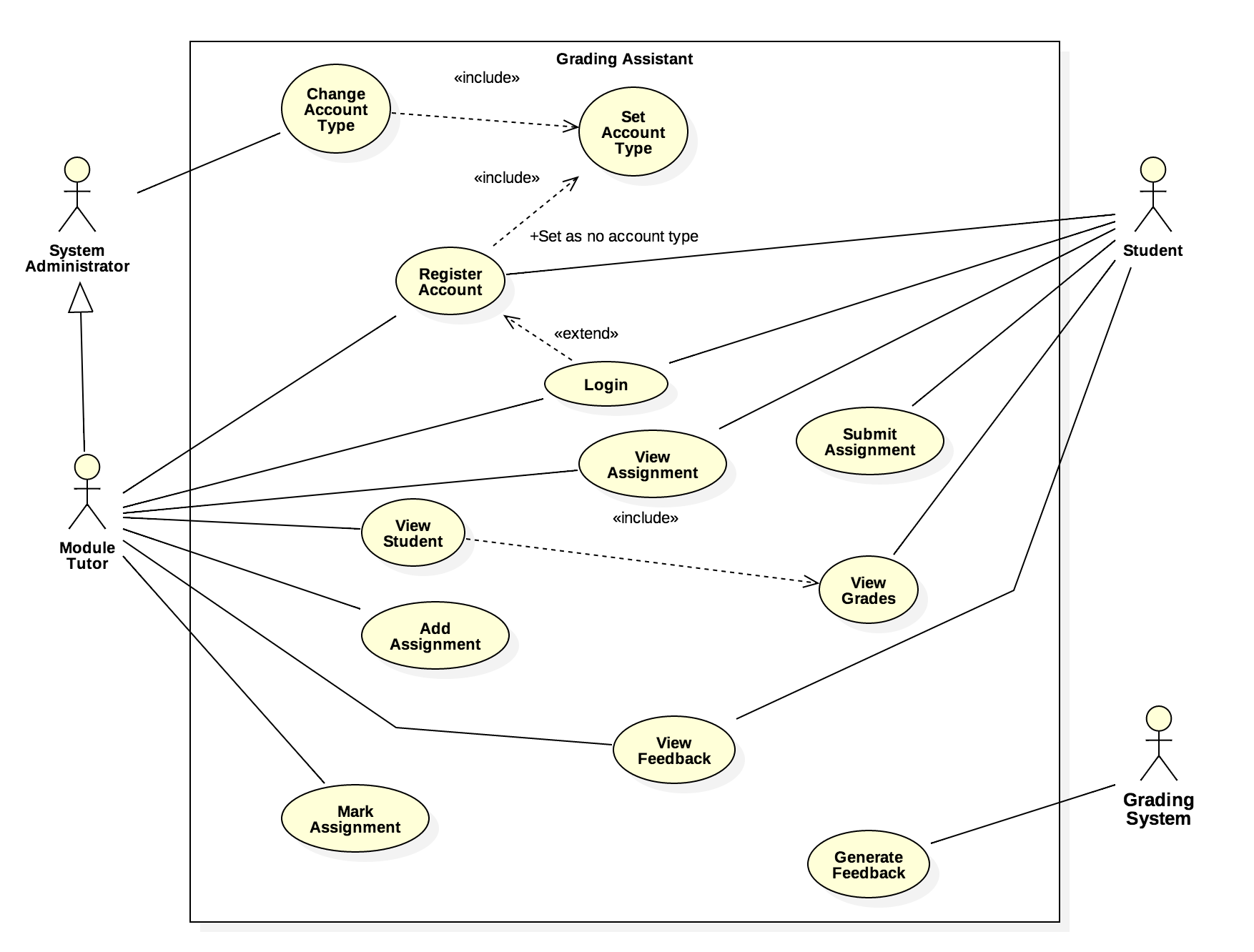
Upguard.com. (2016). *MySQL vs MongoDB*. [online] Available at: https://www.upguard.com/articles/mysql-vs-mongodb [Accessed 24 Nov. 2016].

W3schools.com. (2016). *Browser Statistics*. [online] Available at: http://www.w3schools.com/browsers/ [Accessed 2 Dec. 2016].

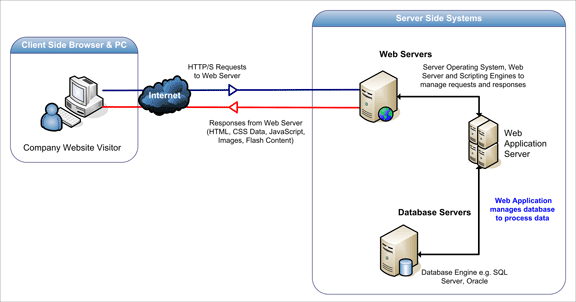
Webhostdesignpost.com. (2016). *Website Storyboarding | Examples, How To and Sitemap*. [online] Available at: http://www.webhostdesignpost.com/website/websitestoryboarding.html [Accessed 24 Nov. 2016].

# Appendices

Appendix A



Appendix B



(Acunetix, 2016)

Appendix C

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2016 | Chrome | IE | FireFox | Safari | Opera |
| October | 73.0% | 5.2% | 15.7% | 3.6% | 1.1% |
| September | 72.5% | 5.3% | 16.3% | 3.5% | 1.0% |
| August | 72.4% | 5.2% | 16.8% | 3.2% | 1.1% |
| July | 71.9% | 5.2% | 17.1% | 3.2% | 1.1% |
| June | 71.7% | 5.6% | 17.0% | 3.3% | 1.1% |
| May | 71.4% | 5.7% | 16.9% | 3.6% | 1.2% |
| April | 70.4% | 5.8% | 17.5% | 3.7% | 1.3% |
| March | 69.9% | 6.1% | 17.8% | 3.6% | 1.3% |
| February | 69.0% | 6.2% | 18.6% | 3.7% | 1.3% |
| January | 68.4% | 6.2% | 18.8% | 3.7% | 1.4% |

(W3schools.com, 2016)